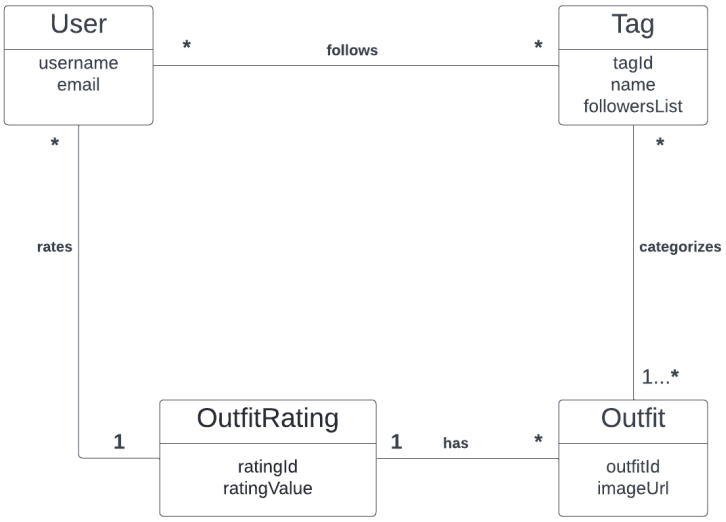
Deliverable – Sprint 3

1. Domain Model



Domain Model remains the same since the newly added components are mostly in the front-end and face detection related configuration.

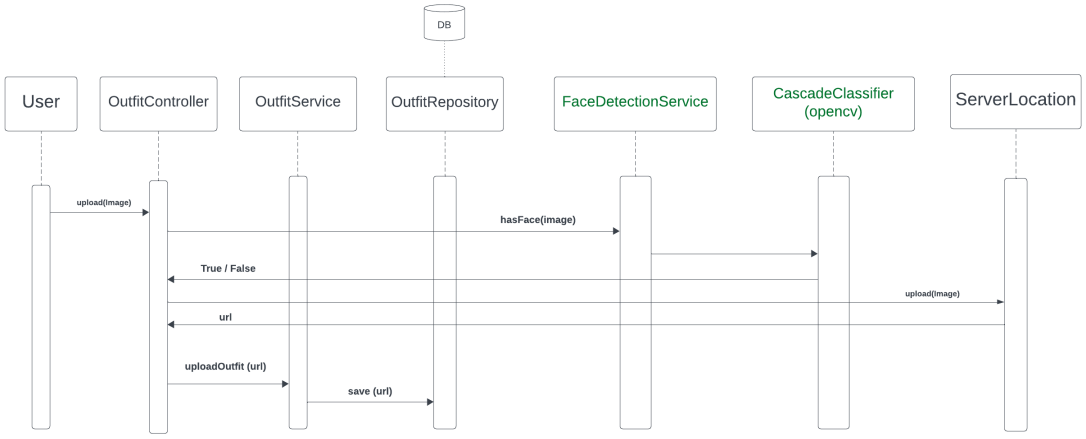
Adding them did not change the essence of the workflow, just built on top of existing blueprint.

2. Interaction Diagrams

Most of the interaction diagrams remain the same from Sprint 2 deliverable outcomes.

2.1 Interaction / Sequence Diagrams – *Upload Outfit*

The updated interaction diagram for uploading outfit is as follows.



old work
new work

A Peek at the Image Detection Flow

- The user first uploads an image via the REST API endpoint.
- Instead of directly uploading the outfit, we make a call to the face detection service.
- OpenCV's **Imgcodecs** class is used to load the uploaded image into memory (**Imgcodecs.imread**).
- The **CascadeClassifier** class in OpenCV is initialized with the Haarcascade XML file (**haarcascade_frontalface_alt.xml**). This file contains the pre-trained model for face detection.
- The **detectMultiScale** method of **CascadeClassifier** is used to detect faces in the image.
- Based on the output of the face detection classifier, we decide whether the user can upload the outfit or not

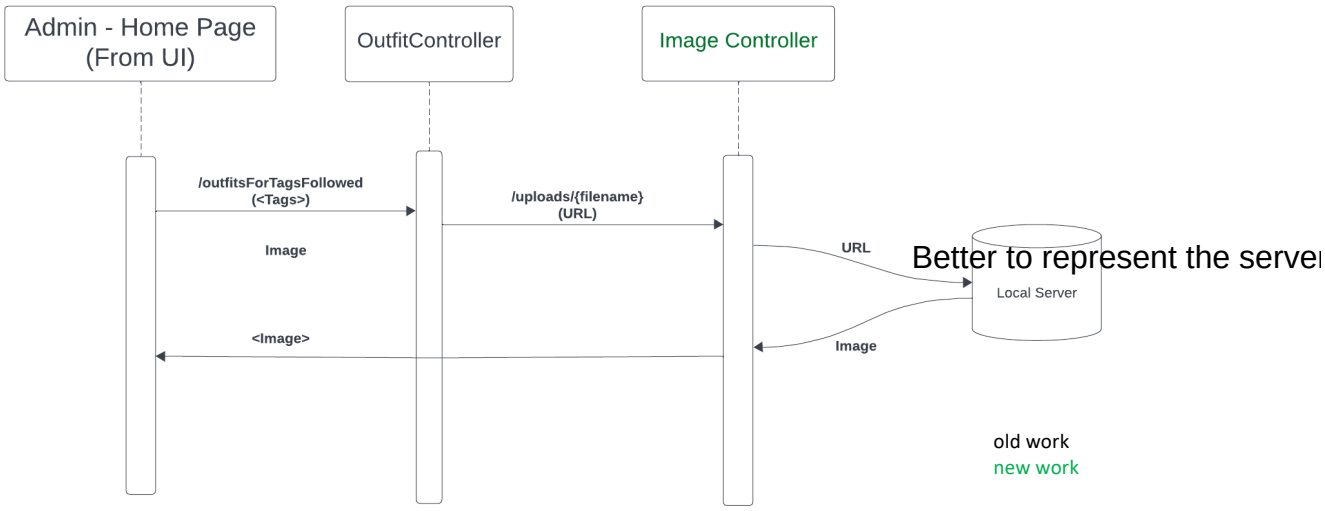
Compilation of OpenCV Jar File:

The OpenCV Jar file with Java bindings was compiled from the OpenCV source code. This process involved configuring OpenCV with CMake, specifying options to enable Java bindings, and compiling the source using a compatible JDK version. The resulting Jar file and the associated native libraries (libopencv_java4xx.dylib or equivalent) are used in the application.

2.2 Interaction / Sequence Diagrams – *Upload Outfit Retrieval Part*

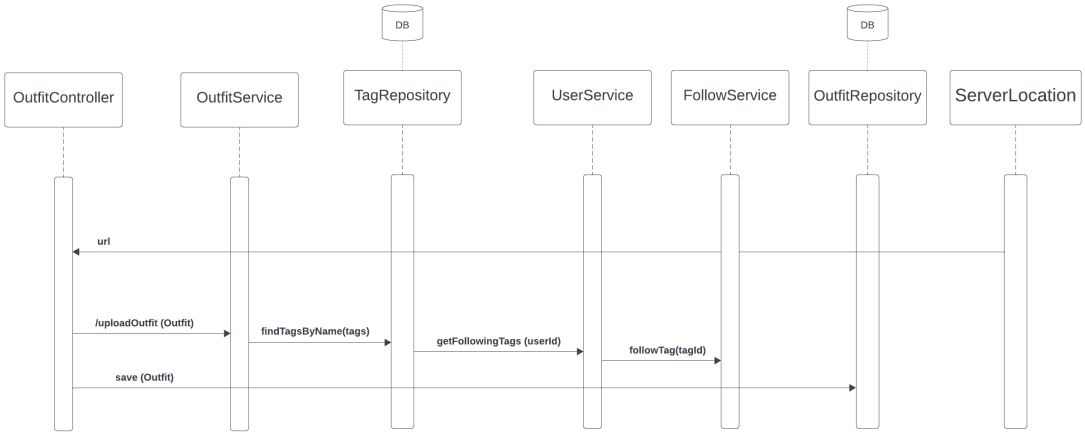
This part was added as part of bug fixing the retrieval of image after uploading from the front-end process.

- After uploading, we redirect the user to home page where all the outfits that he follows are shown
- As part of that we hit the new API written in Image Controller which retrieves from the URL of the image passed

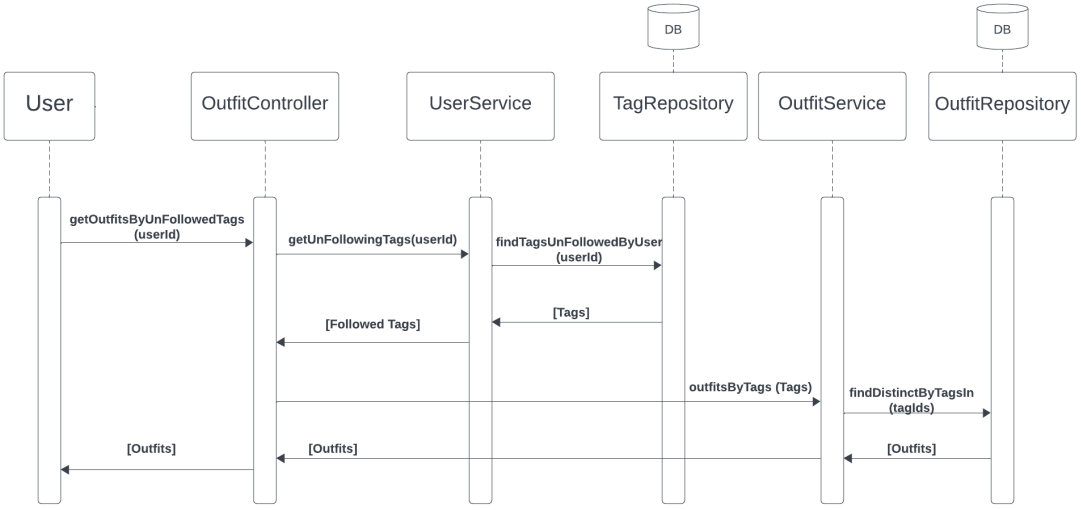


2.3 Interaction / Sequence Diagrams – *Upload Outfit Auto-Follow Tag Part*

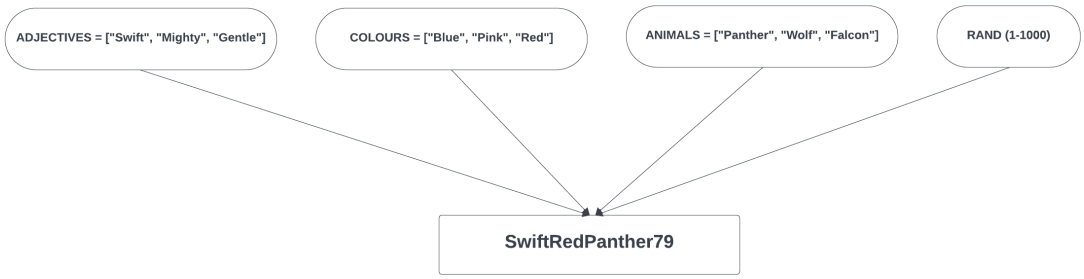
This interaction ensures the user follows the tags that he adds while uploading an outfit.



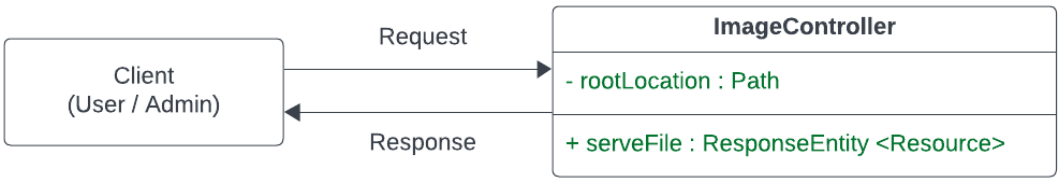
2.4 Interaction / Sequence Diagrams – *Explore Page (outfitsForUnfollowedTags)*



2.5 Anonymous Username generation

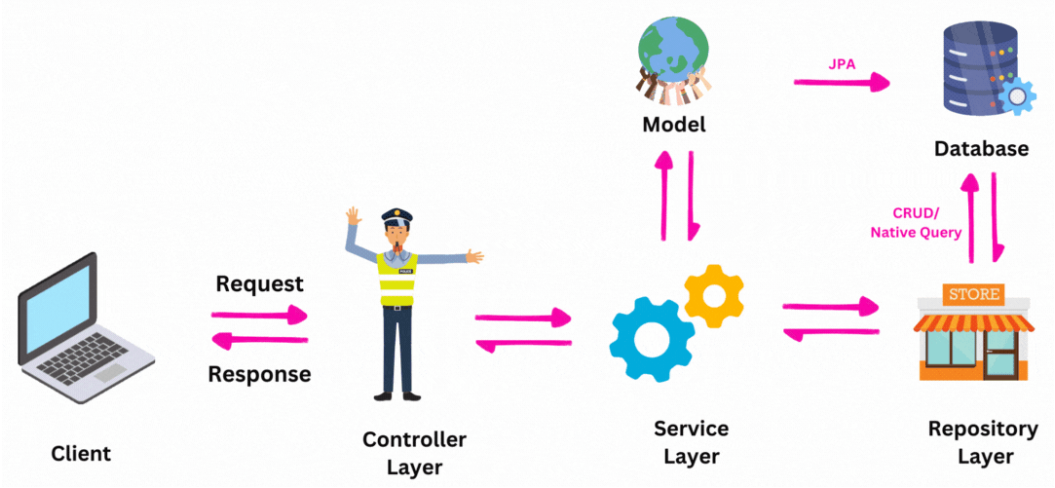


3. Class Diagrams



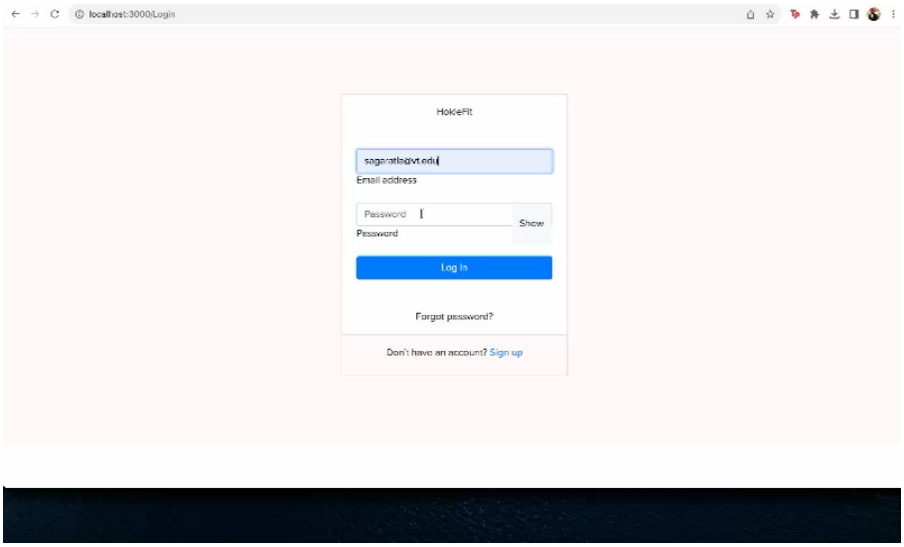
old work
new work

All the other classes remain the same as in previous sprint report. The previous report briefly talks about the Spring Boot Architecture



source

4. Demo



5. Scrum

Please add more screenshots or at least paste a URL of the d

5.1 Sprint 3 Backlog

5.1.1 Face Detection API – OpenCV

- Assignee - Sagar Atla
- Tasks -
    - Research & choose a face detection library or service
        - Estimated Time – 2hrs
    - Implement a Face Detection Service
        - Estimated Time – 2hrs
    - Integrate Face Detection with the Outfit Upload Process
        - Estimate Time – 2hrs
    - UI Integration for Face Detection
        - Estimate Time – 2hrs

5.1.2 Anonymous Username Generation

- Assignee – Sagar Atla
- Tasks -
    - Develop algorithm for generating unique anonymous usernames
        - Estimated Time – 1hr
    - Implement username generation in the account creation process
        - Estimated Time – 1hr

5.1.3 UI – Showing Tags

- Assignee – Srikanth Karri
- Tasks -
    - Design Tag display layout on the UI
        - Estimated Time – 1hr
    - Implement the API to retrieve Tags for each outfit
        - Estimated Time – 1hr
    - Integrate tag display functionality on the UI
        - Estimated Time – 1hr

5.1.4 UI – Fix Upload Outfit Functionality

- Assignee – Mohith Kamanuru
- Estimated Time – 8hrs

5.1.5 Auto-Follow Tags while Uploading Outfits

- Assignee – Mohith Kamanuru
- Estimated Time – 2hrs

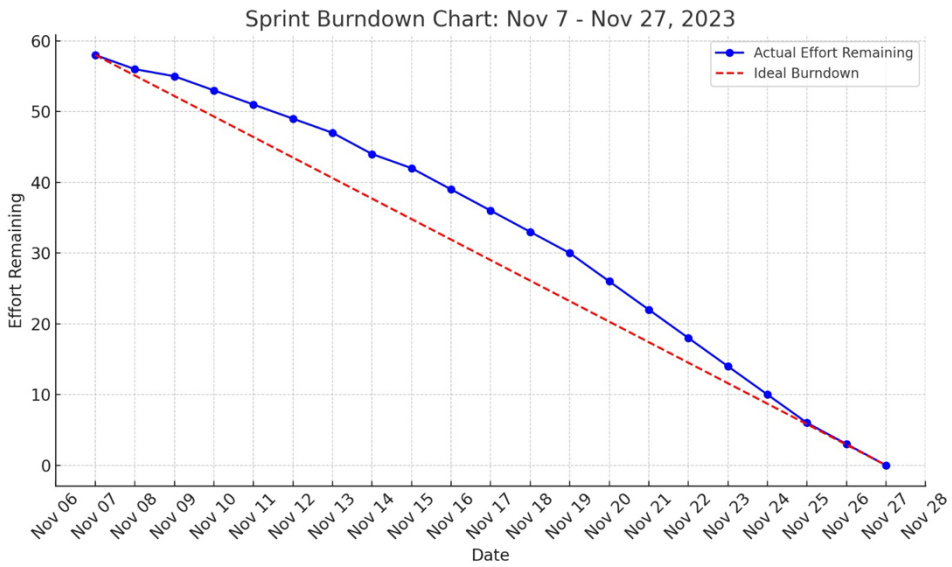
5.1.6 Explore / Search Functionality Page

- Assignee – Srikanth Karri
- Tasks -
    - Design search functionality UI
        - Estimated Time – 4hrs
    - Develop API for searching outfits
        - Estimated Time – 3hrs
    - Integrate search API with frontend
        - Estimated Time – 3hrs

5.1.7 Testing

- Assignee – Aruj Nayak
- Tasks -
    - ~~Unit Testing Backend Functionalities~~
        - ~~Estimated Time – 2hrs~~
    - Dummy Data for Integration Testing
        - Estimated Time – 2hrs
    - Integration testing
        - Estimated Time – 2hrs
    - Usability Testing
        - Estimated Time – 2hrs

5.2 Burndown Chart



5.3 Retrospective

Accomplishments:

- We were able to bring the project to a near completion without any major impediments
- We were able to get OpenCV to work on different teammates' machines with different JVM versions

Impediments:

- Getting OpenCV to work on everyone's machine with JVM version was a herculean task
- Incomplete UI for Explore / Search functionality
- Could not implement Unit tests for the backend functionalities

Addressing Impediments:

- March on to finish the UI for explore / search functionality

What Went Well:

- Team collaboration continued to be a strong point from Sprint 2
- Sprint planning was effective, leading to a clearer understanding of tasks & the estimates were apt

Improvements:

- Could have written unit tests along with backend functionalities which would have avoided writing all of them at the end

5.4 Product Backlog Updated

Newly Identified Stories

- Adding Dummy Data was identified as a sub task that was required as part of Integration Testing. It required us to add a lot of data in different tables

Dropped Stories

- Unit testing of all backend functionalities